IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

APPLICANTS:	Pawan Goyal
SERIAL NO.:	09/503,975
FILING DATE:	February 14, 2000
TITLE:	Restricting Communication of Selected Processes to a Set of Specific Network Addresses
EXAMINER:	Larry D. Donaghue
GROUP ART UNIT:	2154
ATTY. DKT. NO.:	21816-04464

MAIL STOP AMENDMENT
COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450

Declaration of Pawan Goyal

I, Pawan Goyal, hereby declare the following:

1. I am the inventor of the invention described and claimed in United States Patent Application Serial Number 09/503,975, entitled "Restricting Communication of Selected Processes to a Set of Specific Network Addresses," filed on February 14, 2000 (hereafter "the Application").

2. I am providing this declaration to establish that the invention described and claimed in the Application was conceived and reduced to practice prior to the filing dates of 1) United States Patent 6,529,985 to Deianov et al., filed February 4, 2000 ("Deianov") and 2) United States Patent 6,754,716 to Sharma et al., filed February 11, 2000 ("Sharma"). The Examiner cited Deianov and Sharma in an Office Action dated January 26, 2005.

3. I was employed by Ensim Corporation ("Ensim") from January 1999 to January 2002. My title at Ensim was Engineering Director. Ensim has a place of business at 1366 Borregas Avenue, Sunnyvale, CA 94089. Ensim is the assignee of the Application.

4. During my employment with Ensim, I conceived of methods and computer program products for restricting communication of selected processes to a set of specific network addresses (hereafter "the Product") that are now described and claimed in the Application. The Product was reduced to practice by myself and others at Ensim as a software component prior to February 4, 2000, the filing date of Deianov, and February 11, 2000, the filing date of Sharma.

5. Attached hereto as Exhibit A is a redacted version of a true and correct copy of an internal Ensim document describing the Product. This document was prepared using information provided by me to document the operation of the Product.

6. Accordingly, the invention described and claimed by the Application was reduced to practice prior to February 4, 2000.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

2/14/05
Date

Pawan Goyal

Ex. A

1. The fexec component is informed, using a mechanism such as its command line or a batch file, about the application to be run and the interface with which the application should be associated.
2. The fexec module uses standard system calls to determine information about the interface such as its IP address and interface flags and stores this information internally.
3. The fexec component loads an *interception kernel module* that contains within it a translation table into the running kernel.
4. When the interception module is initialized, it overwrites the kernel's system call table so that the *fork, clone,* and *exit* system calls are diverted to entry points in the interception module. It also modifies the TCP protocol stack so that the *bind* and *connect* procedures are also diverted to the interception module.
5. The fexec component communicates with the interception module using any of the standard mechanisms for user-kernel communication and tells the module its process ID and the interface that it obtained in step 2.
6. The interception module stores this information in its translation table in an entry of the form <pid, interface info>
7. Finally, the fexec component uses the *exec* system call to overwrite its binary image with that of the specified application.
8. When the application performs one of the system calls that are listed in step 4, the interception module is called by the kernel.
9. On a *fork* or *clone* system call, the translation table is updated so that the pid created by these calls is also associated with the interface information.
10. On a *bind* system call, the following changes are made before the original bind function is invoked:
    - If the bind is to the IP address that was supposed to be associated with that pid anyway, the call is not modified
    - If the bind is to the 'localhost' special address (127.0.0.1) then the localhost argument in the system call is modified from 127.0.0.1 to the IP address stored in the translation table
    - If the bind is to the 'wildcard special address (INADDR_ANY) then the wildcard argument in the system call is modified from INADDR_ANY to the IP address stored in the translation table
    - A bind to any other address is notified as an error
11. On a *connect* system call the following changes are made before the original connect function is invoked

- If the connect is to the 'localhost' special address (127.0.0.1) then the localhost argument in the system call is modified from 127.0.0.1 to the IP address stored in the translation table
- If the connect is to the 'wildcard' special address (0.0.0.0) then the wildcard argument in the system call is modified from 0.0.0.0 to the IP address stored in the translation table
- A connect to any other address is unmodified

12. On an *exit* system call, the translation table entry is cleared for that pid before the original call is invoked.